Commonly asked Interview Questions

1) What is difference between between Test Scenario and Test case?
2) What is the difference between Smoke Testing & Sanity Testing?
3) Explain Regression Testing and its types?
4) Explain Bug Life Cycle.
5) Explain STLC.
6) Explain RTM(Requirement Traceability Matrix).
7) Explain Test Plan and Test Strategy.
8) Explain meetings involved in Agile?
9) What is Integration Testing. Explain its types.
10) What are the roles of a Software Tester?

Commonly asked Interview Questions

11) Explain Test case Design Technique.

12) What is the difference between Static & Dynamic Testing?

13) What are the levels of Testing?

14) What is the difference between Blackbox Testing and Whitebox Testing?

15) What are the principles of Software Testing?

16) What are the challenges in Software Testing?

17) What is the difference between QA and QC?

18) What is the difference between Verification and Validation?

19) What are the differences between Waterfall model & Agile model?

20) What do you mean by Severity and Priority?

Commonly asked Interview Questions

21) Give an example for High Severity Low Priority and Low severity High Priority?

22) What is a defect?

23) During the initial phase of project, when build is still being developed by Developer. What will Tester do?

24) Write Test cases on Flipkart add to cart, search, payment functionality.

(Any E-commerce application may be asked)

25) Write Test cases on Water bottle, white paper, pen, lift, bike, tooth brush, calculator, laptop.

Terminologies in Agile

Sprint - Actual time taken to develop and test 1 or more user stories

User story - Collection of (small set)requirements(Nothing but features)

Epic - Collection of user stories(complete set of requirement)

Release - Combination of Sprints.

Scrum master - He is responsible for completion of project and get work done from team members.

Meetings/ceremonies in Agile(Interview)

1) Sprint Planning(1.5 hours)
2) Daily Standup or Scrum meeting(15-20 mins)
3) Sprint Retrospect meeting(1 hour)
4) Release Retrospect meeting(1 hour)
5) Bug Triage Meeting
6) Product Backlog
7) Sprint Review

Sprint Planning(1.5 hours)

This is conducted on first day of every sprint. Scrum master will conduct this meeting.

BA priortizes user stories according to customer requirement and explains requirement to the entire team. Scrum master assigns work to developer and Testers based on their availability.

BA, Scrum Master, Developers, Testers will be part of the meeting.

Daily Standup or Scrum meeting(15-20 mins)

This meeting is conducted on daily basis.

Scrum master will conduct this meeting and discuss the plans for the day and asks team if there is any issues, so that they can address it.

BA, Scrum Master, Developers, Testers will be part of the meeting.

Sprint retrospect Meeting

Conducted on last day of the Sprint. It is conducted by Scrum Master.

We discuss what went well, what did not go well and what are the action plans.

BA, Scrum Master, Developers, Testers will be part of the meeting.

Release retrospect Meeting

Conducted on last day of the Release. It is conducted by Scrum Master.

We discuss what went well, what did not go well and what are the action plans.

BA, Scrum Master, Developers, Testers will be part of the meeting.

Bug Triage meeting

It is conducted few days before the sprint.

We prioritize defects here(which needs to be fixed first). It is conducted by Sr Test Engineer. BA, Scrum Master, Developers, Testers will be part of the meeting.

Sprint Review

Demo given to customer/product Owner

Product backlog

It contains complete list of user stories and pending work(user stories and defects)

Advantages of Agile

1) Customer satisfaction can be achieved through quick delivery of working piece of software.
2) Requirement changes are allowed
3) There is good communication between team members and Stake holders
4) Teams are self organized
5) Releases will be short
6) It is a simple model to follow

Disadvantages of Agile

1) Effort estimation cannot be done at the beginning
2) Require experienced resources
3) If requirement changes comes during end of the release, it delays the project duration

Waterfall Model	Agile Model
It is based on traditional process management	It is a project based on cycle
Process flow is linear	Process flow is cyclical
Verification is done after implementation	Verification is done during implementation
Project cost is fixed	Project cost is variable
Requirements are freezed at the beginning	Requirements are expected to change
Hierarchical representation of team	Decision made by entire team members

Types of Testing

Manual and Automation

Types of Manual Testing - Whitebox, Blackbox and Greybox Testing.

Types of Blackbox Testing - Functional and Non-Functional Testing

Blackbox Testing and Whitebox Testing(Interview)

Testing the application when it is already developed and ready for Testing or when code is not visible is called as Black Box Testing. It is also called as Closed box Testing. Testers will do Black box testing.

Looking into the source code and checking whether each and every line of the code is working as expected or not is called as White box testing. It is also called as Open box/Transparent testing. Developer will do White box testing.

Types of Functional Testing:

1) Component Testing
2) Integration Testing
3) System Testing
4) Smoke Testing
5) Sanity Testing
6) Regression Testing
7) User Acceptance Testing

What is Positive and Negative Testing

Positive Testing - Testing the application by entering data according to customer requirement is called as Positive Testing. Or

Testing the application by entering valid data is called as Positive Testing

Negative Testing - Testing the application by entering data against customer requirement is called as Negative Testing. Or

Testing the application by entering invalid data is called as Negative Testing.

Integration Testing (Interview)

Testing the data flow or Interface between two or more modules/features is called as Integration Testing.

Why Integration Testing - To check whether the data flow between the modules is working as per customer requirement specification.

Types of Integration Testing: Incremental and Non-Incremental Integration Testing

Explanation

Incremental Integration Testing - Incrementally adding the module and testing the data flow between the modules is called as Incremental Integration Testing.

Types of Incremental Integration Testing

1) **Topdown approach -** Incrementally adding the modules and testing the data flow between the modules and ensure that the module we are adding is child of the previous module is called Topdown Incremental Integration Testing.
2) **Bottom up approach** - Incrementally adding the modules and testing the dataflow between the modules and ensure that the module we are adding is the parent of previous module is called as Bottom up approach.

Non Incremental Integration Testing - Testing the data flow between multiple modules and check whether the dataflow is happening correctly or not is called as Non Incremental Testing. It is also called as Big bang method.

We go for this Testing when it is difficult to identify which is parent module and child module, when multiple modules are there, when there is shortage of time.

Sandwich Testing - Combination of Topdown and bottom up incremental integration testing is called called sandwich Testing.

Smoke Testing(for Interview)

Testing the basic and critical features of an application before doing thorough Testing is called as Smoke Testing.

Why Smoke Testing - To check whether build is testable or not.

Smoke Testing can also be called as Positive Testing or build verification Testing.

Sanity Testing

Sanity Testing is a deep and narrow Testing. It is usually for done when Tester finds a bug and developer has fixed it, we do sanity testing to make sure that other features are working by Testing thoroughly, we test each and every feature and check data flow and end to end features - this is how we do deep testing.

Sanity Testing is done for stable builds.

Key Differences between Smoke and Sanity Testing

(for interview)

Smoke Testing

1) It is a high level Testing(Testing is done on a high level)
2) Testing is done on unstable builds or new builds
3) Testing can be done by both Developers and Testers
4) Only positive testing can be done
5) As soon as we get a new build, we do Smoke Testing

-

Sanity Testing (for interview)

1) It is a deep and narrow testing
2) Testing is done on stable builds
3) Testing will be done only by Testers
4) Both positive and negative testing can be done
5) Sanity Testing will be done when a bug is fixed.

What is Retesting?(for Interview)

When a Tester finds a bug, he will inform to Developer, developer will fix the bug and give back to tester. Tester needs to check if this bug is really fixed or not. This process is called as Retesting.

Or

Process of checking if the developer has really fixed the bug or not is called Retesting.

Regression Testing(Release Candidate Testing)(Interview)

Testing the old features so that any new changes like adding a feature, modifying a feature, deleting a feature or fixing a defect might introduce a new defect in old features, to check that we do Regression testing on the old features.

(Here we test the new changes first and then we test for old features, these old features are tested by looking into old test cases, which are already used earlier)

Why Regression Testing - To check if new changes made has introduced any defect in old features or not.

Types of Regression Testing(Interview)

Unit Regression Testing - Testing only the changes made is called as Unit Regression testing(Here we test only new feature added, because we are sure it is not affecting old features).

Regional Regression Testing - Testing the changes made and the impacted modules affected because of this change is called as Regional Regression testing.

How do we know which all areas are impacted?

By doing impact analysis meeting, in this meeting we cross verify with Business Analyst and check which all areas(modules) are impacted because of this new change made.

-

Full Regression Testing - If the new changes made is largely impacting other modules(more areas), we go for Full Regression Testing.

Full Regression Testing, takes a lot of time.

Both Manual Testing and Automation Testing can be done for Full Regression testing.

Regression Testing	Retesting
Testing the old features so that any new changes are not introducing any defects in old feature is called as Regression Testing.	Once bug is found and Developer will fix it and give back to Testing. As a Tester, we need to recheck if the bug is really fixed or not. This process is called Retesting.
This testing is done for passed Test cases.	This Testing is done for failed Test cases.
It is a Generic Testing.	It is a planned Testing.
We go for automation.	We don't go for Automation.
It is less priority compared to Retesting.	It is high priority compared to Regression Testing.

Acceptance Testing (User Acceptance Testing)

It is an end to end testing, done by the customers where in they use software for real time business for some particular period of time and check whether software can handle all the real time business, scenarios and situations. This is called as Acceptance Testing.

Why acceptance testing?

1) To ensure that application meets the business requirements.
2) To make sure that software company is not developing wrong features.

Types of Non Functional Testing

Compatibility testing

Performance Testing

Performance Testing(Interview)

Testing the stability and response time of an application by applying load is called as Performance testing.

Why performance testing - To check how is the response time and stability of an application when load is applied.

Stability means checking how the many users can software can accept(ability to withstand designed number of users).

Response Time means time taken by server to respond to user query

Compatibility Testing(Cross browser testing)

Testing the functionality of an application or software for different software configurations is called Compatibility Testing.

To ensure all the features is consistently working for all platforms(different browsers, OS, versions) we will do compatibility testing.

How tester will do compatibility testing

> Initially Testers will test for base platform i,e,. Lets say among the browsers - chrome, firefox, safari, according to market research chrome is widely used, hence we will consider chrome as the base platform and start the software in Chrome first and later we will test for other browsers.

Functional Testing key points

1) It is done before Non-Functional Testing
2) It validates/checks functionality of the software
3) Functionality describes what software does
4) Works on customer requirements

Non Functional Testing key points

1) It is done after Functional Testing
2) It validates/checks non- functionality of the software
3) Functionality describes how software works
4) Works on customer expectations

Test scenarios & Test cases(Interview)

Test scenarios are high level documentation of requirements or high level documentation of the functionalities which are to be tested.

Test cases are detailed description of Test scenarios.

What are the Levels of Testing:(Interview)

Unit Testing - Testing each and and every unit/component of a software.

Integration Testing - Testing the data flow between 2 or 3 modules

System Testing - Here, we do end to end Testing and check whether the end feature is working as expected or not.

Acceptance Testing - End to end testing done on software for a particular period of time and check if it can handle customer business requirement or not. It is done by customers.

Difference between Static & Dynamic Testing(Interview)

Static Testing - Verification is the example, It is done before software is developed. Here we do walkthrough meeting, auditing, We will try to prevent the defects. It is Low cost. We conduct Reviews here.

Dynamic Testing - Validation is the example, It is done after software is developed. It is High cost. We do component Testing, Integration Testing, System Testing etc and will fix the defects.

TEST STRATEGY (for Interview)

Test strategy is a high level plan done at a organisation level.

Test Strategy remains fixed from project to project since it is done at a organisation level

It provides a structured approach to the team to conduct Testing activities.

- Components of Test Strategy - Scope & Overview, Testing methodology, Testing environment, Testing tools, Release, Risk analysis, Review and approvals

Test Plan(for Interview)

It is a document which contains all the future activities of a Test Engineer and testing activities to be conducted.

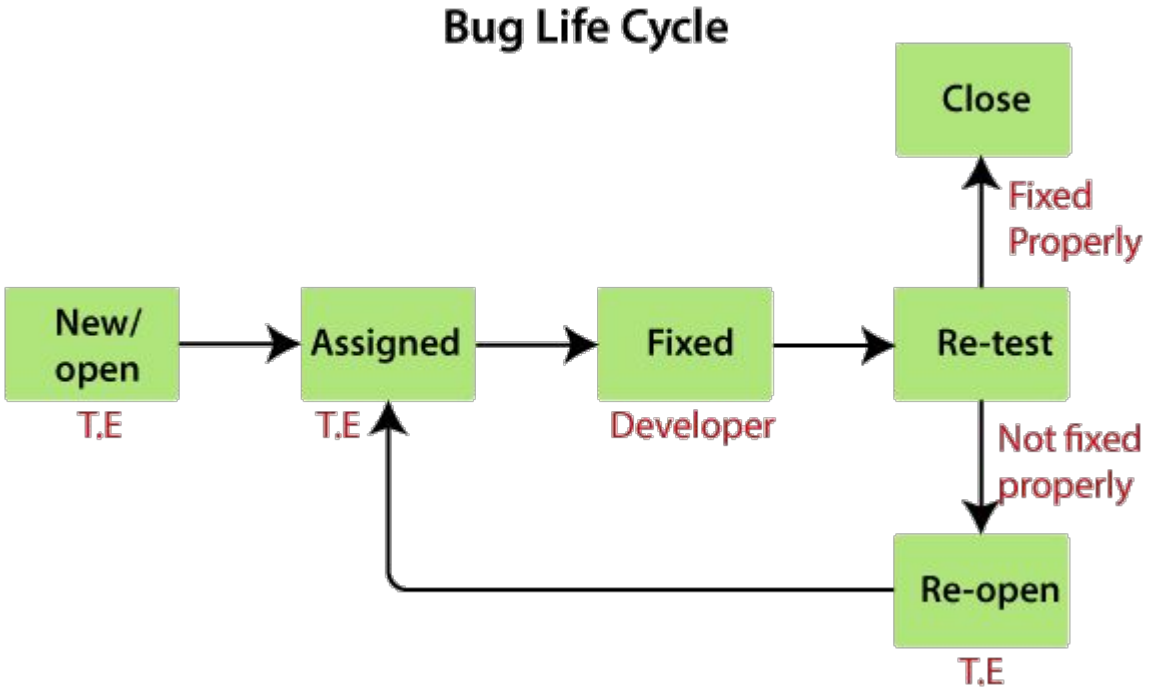
Test Plan is derived from Test Strategy.

Test Plan varies from Project to Project.

Test Plan is prepared by Test Lead

There are 15 attributes in Test Plan, They are - Objective, Scope, Test Methodology, Test approach, Assumption, Risk, Mitigation Plan, Roles of Tester, Scheduling, Defect Tracking, Entry and Exit Criteria, Test Automation, Deliverables, Templates.

Bug Life Cycle



Explanation for Bug/Defect Life Cycle(Interview)

As soon as Test Engineer, finds a defect, he will login to a defect tracking tool(Jira), keeping CC to Test Lead and Development Lead. Initially status of the defect would be “New”, once it is confirmed by Leads, it will be in “Open” status. Now, the defect will be passed on to developer by concerned leads, now the status of the defect will be “Assigned”. If it is really a defect, developer will fix it and change the status to “Fixed”. Now the defect will be passed on to the Test Engineer where he will Retest if the defect is really fixed or not, now the status will be “Retest”. If the defect is fixed, he will close the defect by changing status to “Closed”. If the defect is not fixed, he will change the status to “Re-open” and re-assign the defect to Developer again. Now the cycle, repeats. This is called as Bug Life Cycle.

Once assigned to developer, he can the change status to:

- Duplicate defect
- Defect cannot be fixed
- It works fine for me/Issue not reproducible
- Postpone/Deferred

When can Developer say it is a “Duplicate defect”

1) If the same defect is logged by other Tester
2) If defect is already fixed

When can developer say “Defect cannot be fixed”

1) When cost of fixing the defect is more than the actual defect
2) When technology itself is not supporting(there is a defect in technology/application which can’t be fixed)
3) When there is a minor defect in the root of the product and it is not affecting the customer business

When developer says that “It works fine for me”

1) When there Build mismatch
2) When there is browser, OS or Platform mismatch
3) When defect report prepared is improper

When can Developer “Postpone/Deferred” the defect

1) When it is a minor defect and received during the end of the release.
2) When it is a minor defect and it is not affecting customer business workflow.
3) When it is minor defect and customer does not want the feature in this release.
4) When it is a defect but customer is planning to do lot of requirement changes in the particular feature.

5 Types of Test case Design technique(Interview)

1) Error Guessing

2) Equivalence Class Partitioning

3) Boundary Value Analysis

4) Decision Table Technique

5) State Transition Diagram

1) **Error Guessing** – Here, TE will guess the error and derive more scenarios.

Example: Assume there is an “Amount” text field and the requirement says that it accepts + integer only.

Now we will have to enter only invalid inputs such as -10, abc, 10@ etc and try to guess more errors in this case.

2) **Equivalence Class Partitioning** - Here, we divide the class or value into equal parts and then do the testing. Let's say, there is an "Amount" text field and the requirement says that it can accept numbers between 100 to 500. We can divide this range into equal classes I,e,. 0 to 99, 100 to 200, 200 to 300, 300 to 400, 400 to 500 & 501 to 600. After this try enter any 1 value from each of this class. Example if you enter 150 for the class 100 to 200, need not enter other values in range of 100 to 200, if it is accepting, then test case is pass, if it is not accepting, then test case is fail. Likewise we can only test 7 scenarios/values(which includes 5 positive & 2 negative scenarios) and ensure that our test case coverage is achieved.

3) **Boundary Value Analysis** – It is used to test between range of values. It is a process of testing extreme boundaries of the given component. Lets say, here you need values between the range A to B. We need to test for A,A+1 & A-1 similarly B, B+1 & B-1 . So here we will have 4 positive scenarios and 2 negative scenarios. Since the boundaries are covered, our test case coverage is good and no need to test for other values.

Advantages - Easy to understand, Test cases are less, cost effective(less)

4) **Decision Table Technique** – In this Technique, we check for multiple conditions, combinations and Rule criterias.

Formula = 2^no of conditions= total no of rules or scenarios

This is based on condition, combination and Rules.

Example1:

Requirement- Customer wants to order from Swiggy,

.

Conditions:

New customer - 15%

Repeat customer - 10%

Coupon code - 30%

No of TC = 2^no of conditions = 2^3=8

5) **State Transition Diagram** – This technique is used to check for different screens of a software. It is basically a pictorial representation of the scenarios.

Example: Let's say a person has to withdraw cash from an ATM machine, we can derive 4 scenarios for this.

1) When he enters the correct pin for the first time, he withdraws the cash.

2) When he enters the wrong pin for the first time and the correct pin for the second time, he can withdraw the cash.

3) When he enters the wrong pin for the first and second time and enters the correct pin for the third time, he can withdraw the cash.

4) When he enters the wrong pin for all attempts I,e,. first, second & third attempt, the card gets blocked and he cannot withdraw the cash.

These scenarios customers can write in a pictorial way and present how scenarios we can derive here.

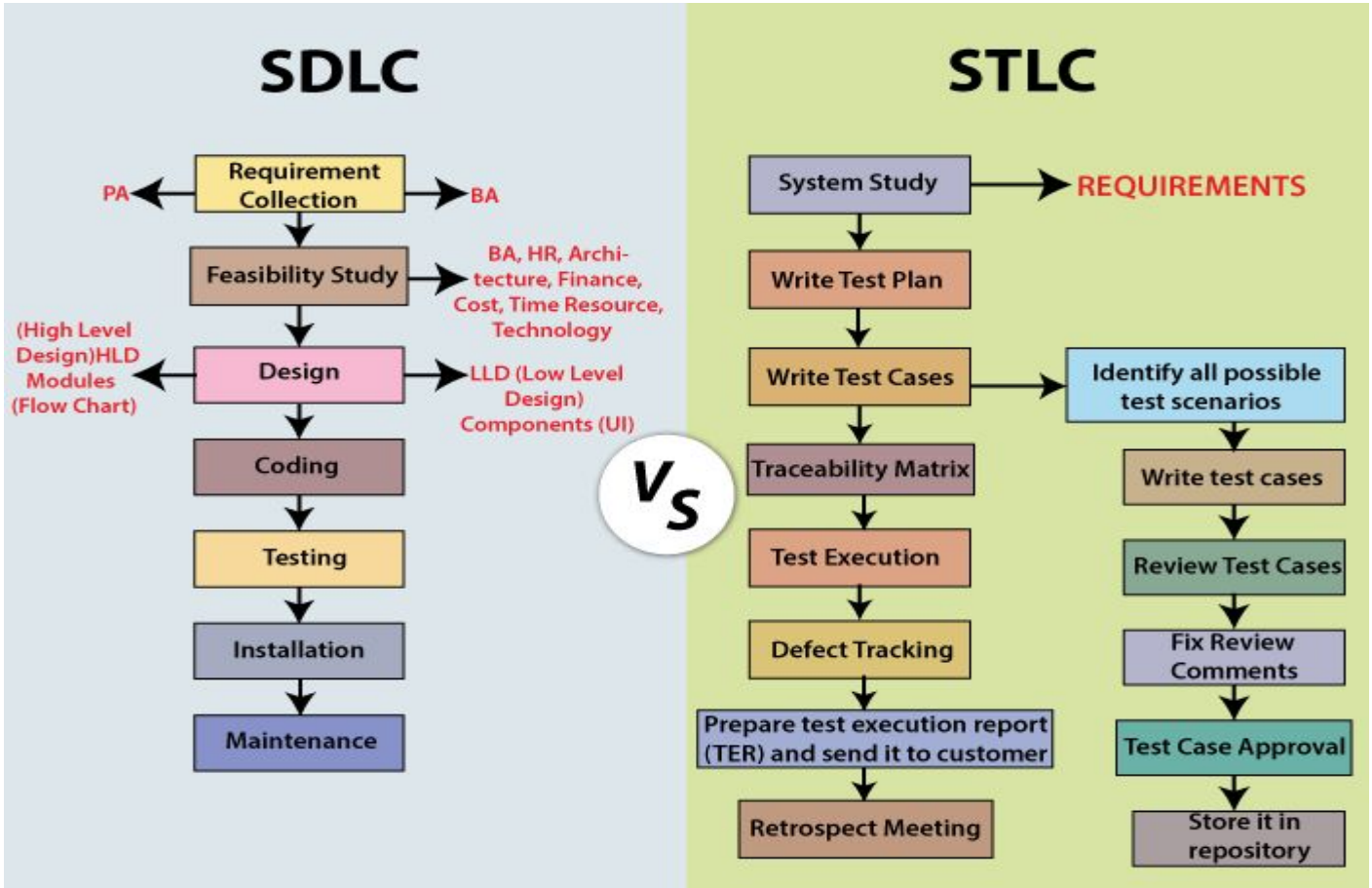
Requirement Traceability matrix (RTM) (Interview)

It is also called as Requirement Traceability matrix(RTM) or Cross reference matrix (CRM). “It main use is to map requirement to Test cases”.

It is a document through which we are ensuring that each and every requirement has minimum test cases.

Types of Requirement traceability matrix:

1) Forward Traceability matrix
2) Backward Traceability matrix
3) Bi-directional Traceability matrix


SDLC
Requirement Collection
PA
BA
Feasibility Study
BA, HR, Archi-tecture, Finance, Cost, Time Resource, Technology
(High Level Design)HLD Modules (Flow Chart)
Design
LLD (Low Level Design) Components (UI)
Coding
Testing
Installation
Maintenance
VS
STLC
System Study
REQUIREMENTS
Write Test Plan
Write Test Cases
Identify all possible test scenarios
Traceability Matrix
Write test cases
Test Execution
Review Test Cases
Defect Tracking
Fix Review Comments
Prepare test execution report (TER) and send it to customer
Test Case Approval
Retrospect Meeting
Store it in repository

STLC(for Interview)

STLC stands for Software Testing Life Cycle, and it represents the series of activities performed during the testing process of a software application or system.

System Study: Customer give requirements in the form of CRS. BA will convert CRS to SRS and BA will explain how each and every feature works for the entire team. BA will also explain customer business scenarios to teammates. Every Test Engineer will understand requirements, while understanding requirements, if they find any issues, they will talk to BA and clarify the doubts. This complete process of understanding the requirements is called System study.

Test Plan: It is a document which derives all the future testing activities. Test plan document consists of information regarding how many TE are required to test in future, which features are to be tested by which Test Engineers, what types of testing to be done on project in future, roles and responsibilities of every Engineer working on project, which Test cases management tool to use, which defect tracking tool to use in future will be maintained in Test plan document.

Test case: Once the Test plan is completed, every TE will start writing Test cases for their assigned modules, where the TE will follow the procedure to write the Test cases. First do System study for assigned modules, later identify all the possible scenarios, document it and conduct brainstorming meetings to have good coverage, later once scenarios are ready, TE will write Test cases by applying Test case design Technique in Test case Template and get it reviewed by reviewers. If there is any problem in Test cases, it will be fixed by TE and will be approved by Test Lead. Finally Test case will be stored Test case repository.

Prepare Traceability matrix – It is a document which ensures that each and every requirement has been mapped to Test cases. There are 3 types of Traceability matrix. They are:

1) **Forward Traceability Matrix** – It is used to map requirements to the test cases. This is done before Test case execution. Here we are ensuring that the product developments are going in the right direction.

2) **Backward Traceability Matrix** – It is used to map Test cases to the requirement. This is done after Test case execution. Here we are ensuring that we are not going against/developing products against customer requirements.

3) **Bi-directional Traceability Matrix** – It is a combination of both Forward Traceability Matrix and Backward Traceability Matrix.

Test case execution: In this stage, developers will give the build to Test Engineers and Test Engineers will start to test the build by executing the Test cases against the application where in TE will do Smoke testing, Functional Testing, Integration Testing, System Testing etc. Since Test Engineers are testing the application by looking into Test cases, it is called Test case execution. Here, we follow the procedure to execute the test cases.

Defect Tracking: While executing the test cases, if the Test engineer finds any defects, he will login to Defect tracking tool and prepare defect report and communicate to developer with unique defect id. Developer will fix the defect and TE will retest the defect and if the defect is really fixed. He will close the bug or TE will re-open the bug. This is called Defect Tracking.

Test case execution report: TL’s will prepare Test case execution report/test case summary report for every build/feature/test cycle/ end of every Release. This report consists of how many Test cases are written, executed, not executed, pass or fail. This report will be sent to Management team, Development team, Testing team, customers.

Release Retrospect meeting or Project closure meeting: Once Testing is completed, at the end of the Project, Test manager will do this meeting called as Release Retrospect meeting where in he will invite all the Test engineers who has worked on project and discuss about the mistakes and achievements done by Test engineers, while working on project.

Note :

Bug Life Cycle is a part of STLC

STLC is a part of SDLC

Principles of Software Testing Interview)

1) We should not do Exhaustive Testing
2) We should do early Testing
3) Testing should be done to identify the bugs in the software
4) We should focus on “Pesticide paradox”
5) We should focus on “Defect clustering”
6) Testing is context dependant(depending on the type of application and customer requirements, testing should be done)
7) Absence of errors in the software does not mean that software is free from defects.

Challenges in Software Testing: (Interview)

1) **Complexity of Software**.
2) **Changing Technology**
3) **Regression Testing**
4) **Incomplete Requirements**
5) **Time Constraints**
6) **Mis -communication Issues**

How to ensure your test case coverage is good?(Interview)

1) By applying Test case design Technique
2) By getting test cases reviewed
3) By writing test cases in Test case template
4) By writing test cases in a simple language
5) By avoiding duplicate test cases
6) By Traceability matrix
7) By conducting brainstorming meeting

Define Severity and Priority?(Interview)

Impact of defect on customer business workflow is called as Severity.

Importance given to fix the defect is called as Priority.

Example of High Severity and High Priority

While logging into login page, blank page is displayed. It is a blocker bug and Priority is P0 because this has large impact on customer business work flow.

Example of High Severity and Low Priority

When user click on Terms and conditions page, blank page is displayed. It is High Severity low priority because this does not affect customer business work flow.

Example of Low Severity and High Priority

When logging in to the login page, lets say login spelling is wrong (logone), now this is low severity, but it will impact the user as it is displayed in the first page is itself. User experience is not good, so it can be considered as Low severity and high priority.

Example of Low Severity and Low Priority

Let's say there is spelling mistake inside contact information page with respect to email Id, phone etc - i,e,. Instead of Email ID it is EMAILID or or any color formation issue or alignment issues. This can be considered as a Low severity and low priority.

What is bug, defects, errors & Failure? (Interview)

Error - It occurs when there is programming mistake in the code that prevents the program from executing or compiling.

Defect - It is a variation between actual result and expected result. It is typically detected when the product enters production.

Bug - It is detected during Testing phase when there is a mismatch. This affects the functionality and performance of the software.

Failure - It is an issue which occurs in Production(If software gets crashed in production, we can term it as Failure)

What are qualities of good test cases?(Interview)

1) Test case coverage should be good.
2) Test case should be with less number of steps.
3) Test case should be simple to understand.
4) Test case should be written by applying Test case Design Technique.
5) Test case should be written in Test case Template.
6) Test case should consist of both positive and negative scenarios.
7) Test case should not have any duplicates.
8) Test case should be able to catch defects.
9) Test case should be easy to convert into automation script.
10) If test cases are given to a new TE, they should be able to understand and execute without hesitations.

What are the Roles of Software Tester?(Interview)

- Understand customer requirement
- Write Test scenarios & Test cases
- Conduct Smoke, Functional, Integration, System, Regression testing etc
- Execute Test cases
- Logging the defect into the defect tracking tool
- Giving KT (Knowledge Transfer) to other team members
- Conduct Brainstorming meeting for better test case coverage & getting test cases reviewed

Difference between Verification & Validation?(Interview)

Verifying if CRS, SRS, HLD and LLD are according to the customer requirements or not is called as Verification. It is done before Software is developed.

The process of testing the functionality of an application is called as Validation. It is done once software is developed.

Note

Key attribute of Test Plan is “Entry and Exit criteria”

Key attribute of Test case template is “Expected Result”

Key attribute of Defect Report Template is “Attached Screenshot”